3P1-IOS-2a-3

Extracting Definitions of Mathematical Expressions in Scientific Papers

Giovanni Yoko Kristianto[*1] Minh-Quoc Nghiem[*2] Yuichiroh Matsubayashi[*3] Akiko Aizawa[*13]

[*1] The University of Tokyo [*2] The Graduate University for Advanced Studies
[*3] National Institute of Informatics

Natural language definitions of mathematical expressions are essential for understanding the mathematical content of scientific papers. A textual description corresponding to a mathematical expression determines the type of symbol or function and the specific name for reference. Our objective is to create an automatic way of extracting definitions of mathematical expressions. We needed to create an annotated corpus since there was no annotated data available on relations between mathematical expressions and their definitions and such annotated data would enable us to compare different approaches to the relation extraction task. This paper introduces guidelines for annotating definitions of mathematical expressions. By using 14 manually annotated papers from Springer, we investigated pattern matching and machine learning based methods in comparison with naive practice based on the nearest noun of the preceding text. The result shows potential of our approach in detecting definitions and the usefulness of our annotated data.

1. Introduction

Mathematics is a science with wide application in various fields, such as physics, astronomy, and computer science. Mathematical knowledge is presented in the form of mathematical expressions or formulas in the scientific papers of many fields. To understand the idea and content of a scientific paper, readers certainly have to know and understand the definitions and meanings of mathematical expressions contained in that paper.

The definitions of mathematical expressions can be found from the natural language text surrounding them. The text provides more information and can help in disambiguating mathematical expressions. Therefore, capturing relations between mathematical expressions and their definitions in a scientific paper is a first step toward understanding mathematical content. Furthermore, these captured relations are useful for improving mathematical information retrieval.

Although an annotated corpus is mandatory for developing and examining methods of extracting such relations, there is no available annotated data containing relations between mathematical expressions and their definitions that we can use for our research. One of the challenges is that it is not always easy to identify whether a phrase of a sentence is a definition of mathematical expression or not. To deal with this problem, we tried to develop a guideline for annotating mathematical expressions and their definitions and develop golden data by using this guideline. Such an annotated data set would enable us compare different approaches to this relation extraction task.

We developed pattern matching and machine learning based methods for extracting the definitions of mathematical expressions. These methods were compared with the naïve practice of basing a definition on the nearest noun of the preceding text.

This paper presents two contributions. First, it gives a guideline for annotating definitions of mathematical expressions. By using this guideline, we can construct a data set containing relations between mathematical expressions and their definitions. The annotation schemes in the guideline are developed in such a way so that it enables any person to do the annotation tasks while the quality of the result is maintained. The second contribution is a description of the implementation of up-to-date information extraction techniques, i.e., pattern matching and machine learning based methods, to extract the definitions of mathematical expressions. We investigated the advantages and disadvantages of each method. Since the main extraction target of our system is the most informative definition, we also reveal the challenges faced in extracting them. This paper attempts to establish a framework for a new information retrieval task that is dedicated to the mathematics domain.

The rest of the paper is organized as follows. Section 2 reviews literature related with our research. Section 3 is an overview of the research. Section 4 explains the process in constructing the dataset. The guideline for annotating this data set and some examples are described in section 5. Our method of extracting definitions is explained in section 6. In section 7, we show the results of our methods on our test corpus. Finally, the last section contains the conclusion of this study and suggestions for improvement.

2. Related Work

There are several researches related to the understanding of mathematical discourse. [Kohlhase 09] proposed the Open Markup Format for Mathematical Document (OMDoc) and a data model for mathematical documents. OMDoc is concerned about the markup of a mathematical document and its statements in relation to the theoretical level of the document. [Zinn 04] questions whether it is possible to build a program that understands mathematical discourse and automatically verifies the correctness of mathematical arguments. As a result, [Zinn 04] proposed a general framework for a mathematical proof engine.

[Cramer 08] determined the principal structures and types of definitions in general language. This study can be used as initial knowledge for developing an annotation guideline for definitions of mathematical expressions, although it is not especially intended for mathematical documents or scientific papers.

Contact: Giovanni Yoko Kristianto, Department of Computer Science, The University of Tokyo, giovanni@nii.ac.jp

[Wolska 04] investigated the semantic relations that build the linguistic meaning of a mathematical text. Research done by [Yokoi 11] attempted to extract descriptions of mathematical expressions from Japanese scientific papers. However, [Yokoi 11] only took the last compound nouns in the phrases as descriptions instead of the complete noun phrases. [Minh-Quoc 10] used CDF (Concept-Description-Formula) in defining coreference relations between formulas and text. Using Wikipedia articles as target data, [Minh-Quoc 10] proposed surface text-based matching and pattern matching for mining coreference relations.

[Jeschke 07] proposed a framework to do mathematical ontology extraction from mathematical texts by means of natural language processing techniques. However, [Jeschke 07] have not dealt with the syntactical analysis of equations and symbols.

3. Our Approach

We used scientific papers as our target dataset, and we limited our focus to LaTeX-formatted papers.

The general steps conducted in our research are shown in Figure 1. First, we convert these papers into XHTML files by using LateXML ([LateXML]). Next, after normalizing (removing HTML tags) and pre-processing, we obtain several text files that represent the papers. The annotation process is applied to these text files.

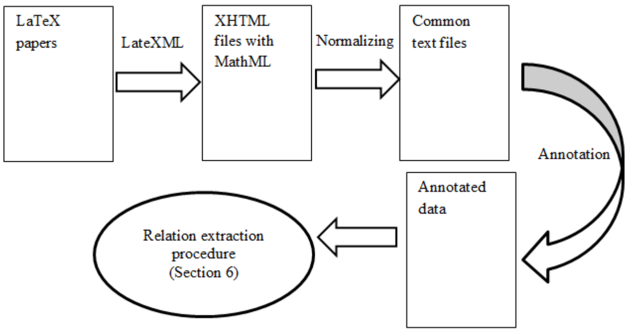


Figure 1. General steps of research

Descriptions of mathematical expressions can take many forms. [Trzeciak 95] distinguishes phrases used in mathematical texts that might contain mathematical descriptions into categories of definition, notation, property, assumption, condition, convention, theorem, and proof. By referring to [Trzeciak 95] and investigating sample papers, we classified the descriptions into definition, property, and value-assignment. We considered that the definition of a mathematical expression would be more or less fixed, whereas the property or value would likely change over the course of the paper. Therefore, we annotated only the definition and distinguished it from other forms of description.

The final step of our research was to develop math-definition extraction methods and compare their performance using annotated data. The evaluation experiment tested three methods: baseline, pattern matching, and machine learning. The baseline method used a naïve practice in which the nearest noun of the preceding text is a definition candidate. The machine learning method extracted all of the noun phrases in the sentence containing the mathematical expression. These noun phrases were then candidates of the definition. We used Conditional Random Field (CRF) with basic pattern features and linguistic features for these noun phrases. The results obtained from machine learning shows that there is potential in our approach to detecting mathematical expression definitions.

4. Construction of the Dataset

There was no source of annotated data that contains relations between mathematical expressions and their definitions, so we had to construct a set ourselves. The steps carried out in this construction are shown in Figure 2.

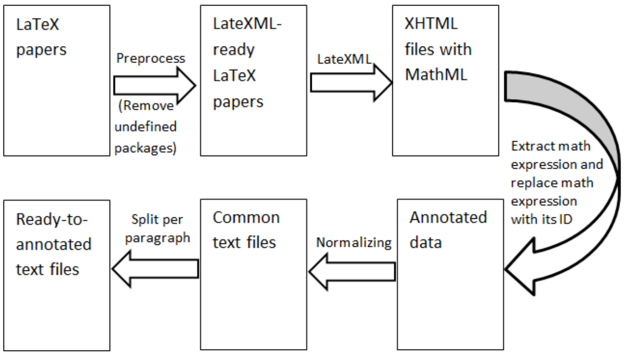

LaTeX papers
Preprocess
(Remove undefined packages)
LateXML-ready LaTeX papers
LateXML
XHTML files with MathML
Extract math expression and replace math expression with its ID
Annotated data
Normalizing
Common text files
Split per paragraph
Ready-to-annotated text files

Figure 2. Dataset construction

Papers used in the dataset must contain enough meaningful mathematical expressions, the content of the paper must be able to be understood, and the relation between each mathematical expression and its definition must be able to be annotated. From these criteria, we choose 14 papers from the International Journal on Document Analysis and Recognition (IJDAR) provided by Springer. These papers are LaTeX-formatted files and can be converted into XHTML files by using LateXML. However, there was a challenge in converting these LaTeX files to XHTML files. Not all external files and packages used in these LaTeX papers have corresponding LateXML macros or bindings to emulate their behavior. In response to this challenge, we removed the reference to these external files and packages as long as the main content of the papers was not disrupted. The final output from the LateXML was XHTML files with mathematical expression expressed in MathML Presentation format ([LateXMLPresentation]).

We gave an ID to every mathematical expression found in XHTML-formatted papers and saved the information (ID, MathML Presentation representation, and Latex representation) regarding these mathematical expressions in database. Later, we replaced all the mathematical expressions in the XHTML files with symbols formatted in MATH_*PaperID_MathID.*

The next step in the dataset construction was normalizing the XHTML files by removing all the HTML tags in them. After that, we obtained papers formatted as a clean text file. We then split the clean text file into several text files. These text files, each representing one paragraph, were ready-to-annotate data.

5. Annotation Design

We developed an annotation guideline that would help us in annotating the definitions of mathematical expressions in scientific papers. This guideline describes how to identify definitions and annotate them.

The principle that we followed is that a definition must be a noun phrase. We considered the noun phrase containing the most information to be the definition. We also have to distinguish definitions from other forms of description.

Table 1. Example of definition annotation

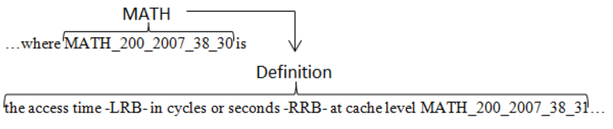
….where α_i is the access time (in cycles or seconds) at cache level i,…
…where MATH_200_2007_38_30 is the access time -LRB- in cycles or seconds -RRB- at cache level MATH_200_2007_38_31 …
 MATH …where MATH_200_2007_38_30 is Definition the access time -LRB- in cycles or seconds -RRB- at cache level MATH_200_2007_38_31…

Table 2. Example of non-definition description

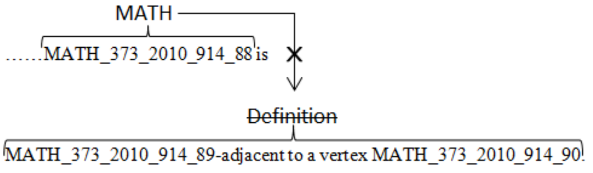
…. v is 1-adjacent to a vertex w.
……MATH_373_2010_914_88 is MATH_373_2010_914_89-adjacent to a vertex MATH_373_2010_914_90.
 MATH ……MATH_373_2010_914_88 is ~~Definition~~ MATH_373_2010_914_89-adjacent to a vertex MATH_373_2010_914_90

Table 1 shows an example of a mathematical expression's definition and Table 2 shows an example of a non-definition description. Table 1 and Table 2 focus only on definitions of mathematical expression α_i (MATH_200_2007_38_30) and v (MATH_373_2010_914_88), respectively. Therefore, information and annotation related to the other mathematical expressions are ignored. The first row of the tables shows the raw text, the second one shows the ready-to-annotate text, and the third one shows the relation between the mathematical expression and its definition candidate. The description "*the access time -LRB- in cycles or seconds -RRB- at cache level MATH_200_2007_38_31*" of MATH_200_2007_38_30 can be easily determined as a definition. The description "*MATH_373_2010_914_89-adjacent to a vertex MATH_373_2010_914_90*" of MATH_373_2010_914_88 also seems to be a definition, although actually it is not, but a characteristic of MATH_373_2010_914_88. Thus, this description is not annotated as a definition.

5.1 Annotation Scheme

The mathematical expression is symbolized in the format of MATH_*PaperID_MathID*. Under these conditions, the mathematical expression will be considered to be a noun by the parser.

Annotation is done by tagging the mathematical expression with the <*math*> tag and the definition of the math expression with the <*definition*> tag. In some cases, there is a possibility of a fragmented definition. To support this sort of definition, we introduce the <*cont-def*> tag. On the other hand, if the truth of the definition is conditional, information in the conditional clause will also be annotated by using the <*assumption*> tag. The list of attributes able to be included in each tag is shown in Table 3.

Table 3. Attributes of Annotation Tags

Attributes of <*math*>	**Value**
id (compulsory)	auto-incremental integer (local value and unique only in inside of annotated text file)
Attributes of <*definition*>	**Value**
id (compulsory)	Format is LocalMathID:DefinitionID
statement (optional)	Default value is "default"
conditional (optional)	Boolean ("false" if not conditional definition and "true" if conditional definition). Default value is "false".
Attributes of <*cont-def*>	**Value**
id (compulsory)	Identifier of parent <definition> tag.
Attributes of <*assumption*>	**Value**
id (compulsory)	Format is LocalMathID:DefinitionID

In the <*definition*> tag, there is a "statement" attribute, which is used to add necessary information about the truth of the annotated definitions. We created three types of definition by specifying three possible values for the "statement" attribute, namely *default*, *notproven*, and *assumption*. The explanation of these possible values is presented in Table 4.

Table 4. Possible values of "statement" attributes

Value of "statement"	**Explanation**
default	The truth of the definition has been proven or the statement where the definition is found is a proposition.
notproven	The truth of definition has not been proven.
assumption	The existence of a math expression and its definition is an assumption. The hint is the presence of keywords *let*, *suppose*, *assume*, or *if* inside the statement.

5.2 Annotation Case Study

In this subsection, we show several most-frequently-encountered construction formats of definitions of mathematical expressions.

(1) Nearest Definition

The nearest definition appears just before the mathematical expression. In some cases, this definition and the target expression together form a noun phrase with the expression as a head noun. An example is the definition of mathematical expression *j* (MATH_200_2007_38_63) presented in Table 5.

Table 5. Example of Nearest Definition

Then for each cache block *j* we load…
Then for each cache block MATH_200_2007_38_63 we load…
Then for each <definition id="3:1">cache block</definition> <math id="3">MATH_200_2007_38_63</math> we load…

(2) Definition using Definitor

A sentence that defines a mathematical expression consists of three parts, i.e., definiendum (mathematical expression), definiens (definition), and definitor (relator verb). The construction format of the sentence can be definiendum-definitor-definiens or definiens-definitor-definiendum. An example is the definition for *A* (MATH_373_2010_924_66) shown in Table 6. This example uses the definiens-definitor-definiendum construction format.

Table 6. Example of Definition in Definiens-Definitor-Definiendum

The adjacency matrix of *Γ* will be denoted by *A*
The adjacency matrix of MATH_373_2010_924_65 will be denoted by MATH_373_2010_924_66
<definition id=14:1>The adjacency matrix of <math id=13> MATH_373_2010_924_65 </math></definition> will be denoted by <math id=14> MATH_373_2010_924_66</math>.

(3) Fragmented Definition

On some occasions, a definition of a mathematical expression might appear in two or more parts separated by a conjunction or other words. An example of this is the definition for the mathematical expression MATH_10032_2006_34_27 shown in Table 7.

Table 7. Example of Fragmented Definition

…with *R* and *T* respectively the radial and angular resolutions
…with MATH_10032_2006_34_27 and MATH_10032_2006_34_28 respectively the radial and angular resolutions
…with <math id=8>MATH_10032_2006_34_27</math> and <math id=9>MATH_10032_2006_34_28</math> respectively <definition id=8:1>the radial</ definition > and < definition id=9:1>angular <cont-def id=8:1>>resolutions</cont-def></ definition >

(4) Conditional Definition

There are several surface forms of a conditional sentence, such as <if A, then B>, <A implies B>, <since A, B>, and <if A, then B, where C>. In annotating the conditional definition, we also annotate the conditional clause (*if*-clause). An example is the definition of W_{ICA} (MATH_10032_2006_28_91) shown in Table 8.

Table 8. Example of Conditional Definition

When $M=N$, matrix W_{ICA} is an estimate of A^{-1},…
When MATH_10032_2006_28_90 , matrix MATH_10032_2006_28_91 is an estimate of MATH_10032_2006_28_92 ,…
When <assumption id="1:2"><math id="0">MATH_10032_2006_28_90</math></assumption> , <description id="1:1">matrix</description> <math id="1">MATH_10032_2006_28_91</math> is <description id="1:2">an estimate of <math id="2">MATH_10032_2006_28_92</math></description> , …

6. Definition Extraction Methods

We suggest that the definitions of mathematical expressions are always noun phrases. A noun phrase can range from a simple one that consist only of a compound noun to a complex one that can consist of a compound noun with a clause, prepositional phrase, *wh*-adverb phrase, or other phrase.

We propose three methods to detect and extract the definitions of mathematical expressions automatically: nearest noun, pattern matching, and machine learning. We chose the nearest-noun method as our baseline.

6.1 Nearest Noun Based Method (Baseline)

In this method, we define the definition as a combination of adjectives and nouns in the text that precedes the target mathematical expression. In some cases, the determiner is also included in that combination. By using this approach, all mathematical expressions will only have one definition that will be only a compound noun without additional phrases. Table 9 provides an example of this baseline method. In this example, the baseline method tries to detect the definition of MATH_200_2008_70_131 by using the preceding text of the expression, that is “In other words, the bijection”. This method will detect “the bijection” as the definition since there is already a determiner in this definition and the text preceding the determiner is a punctuation mark. The POS tag of the detected definition is “DT NN”.

Table 9. Example of Baseline Method

In other words, the bijection σ normalizes G in…
In other words, the bijection MATH_200_2008_70_131 normalizes MATH_200_2008_70_132…

6.2 Pattern Matching Based Method

We presume that there are several sentence patterns used in constructing math-related sentences. In this approach, we use some of sentence patterns introduced by [Trzeciak 95]. We also add other sentence patterns that are frequently used in Graphs and Combinatorics papers from Springer.

There are seven sentence patterns in total in the pattern matching method. Table 10 lists these sentence patterns. MATH, DEF, and OTHERMATH symbols denote the target mathematical expression, its definition, and other mathematical expressions, respectively.

Table 10. List of Sentence Patters

No.	**Sentence Pattern**
1	… denoted (as|by) MATH DEF
2	(let|set) MATH (denote|denotes|be) DEF

3	DEF (is|are)? (denoted|defined|given) (as|by) MATH
4	MATH (denotes|denote|(stand|stands) for|mean|means) DEF
5	MATH (is|are) DEF
6	DEF (is|are) MATH
7	DEF (OTHERMATH)* MATH

There are some cases where the nearest noun definition not only defines a mathematical expression that appears right after it, but also other mathematical expressions that appear after the first mathematical expression. This case often occurs when a paper's author wants to enlist or mention several mathematical variables at the same time. Because of that, we propose pattern 7, which is actually an extension of the nearest noun method.

6.3 Machine Learning Based Method

In the machine learning approach, we first take the parse tree of the sentence containing a mathematical expression. Based on that parse tee information, we extract all the noun phrases in the sentence. These noun phrases are the definition candidates of the mathematical expression. Then, we compare these definition candidates with the golden data (definition obtained from annotation step) to check the validity of the definition candidate. Table 11 shows the features applied to the definition candidates.

Table 11. List of Machine Learning Features

No.	Feature
1	Test whether the sentence matches one of 7 sentence patterns used in pattern matching
2	Test for the existence of colons, commas, or other mathematical expressions between the target mathematical expression and the definition candidate
3	Test whether definition candidate is inside parentheses and mathematical expression is outside parentheses
4	Test how far definition candidates are from target mathematical expression (amount of words between them)
5	Test how closely the definition candidate is located to target mathematical expression (after or before)
6	Surface text and POS tag of two previous and subsequent words of definition candidate
7	Surface text and POS tag of two previous and subsequent words of target mathematical expression
8	Apply unigram, bigram, and trigram (surface text and POS tag) to the beginning and end of the definition candidate
9	Apply unigram and bigram (surface text and POS tag) to the beginning and end of the target mathematical expression
10	Surface text of verb that first appeared between the target mathematical expression and definition candidate

By including the seven sentence patterns used in the pattern matching based method into the list of features, we expect that the performance of the machine learning based method will never be lower than the performance of the pattern matching based method. Feature 2 checks whether there is mention of a mathematical expression in that sentence. Features 3, 4, and 5 check the location of the definition candidate in relation to the target mathematical expression. Feature 6, 7, 8, and 9 give information about the surface texts and POS tags and their combinations surrounding the definition candidate and target mathematical expression. In some cases, the relation between definition and the target mathematical expression can be checked through the verb connecting them. Thus, we represent this information as feature 10.

The machine learning based method is performed using CRFSuite [Okazaki 07].

7. Experiments

7.1 Data Statistics

The experiment was performed on 14 papers. First, we selected ten papers as training data, two papers as development data, and two papers as test data. We used development data to find the best combination of feature sets and applied the machine learning based method with this feature set to the test data.

We extracted continuous definitions that exist in the same sentence with the target mathematical expression. We did not consider extracting fragmented definitions or definitions in different sentences from the target mathematical expression. Table 12 shows the statistics of mathematical expressions in the data set, and Table 13 shows the statistics of the data set used in the 6-fold cross-validation experiment.

Table 12. Data Statistics – Number of mathematical expression that have definition(s)

Category	Number of mathematical expressions with definition(s)
All definitions	558
Continuous definitions in same sentence	540
Definitions softly-detected as noun phrase	422
Definitions strictly-detected as noun phrase	372

Table 13. Data Statistics for 6-fold cross-validation – Number of mathematical expressions with definition(s) detected as noun phrases

Iteration	Training		Test	
	Strict	Soft	Strict	Soft
1	228	258	82	96
2	256	288	54	66
3	256	297	54	57
4	278	319	32	35
5	264	302	46	52
6	268	306	42	48

During the annotation step, we assumed the definitions must be a noun phrase. However, the data statistics in Table 12 show that not every definition is detected as a noun phrase, i.e., 422 out of 540 in soft detection and 372 out of 540 in strict detection.

Therefore, the actual number of definitions used by the machine learning based method was fewer than the initial number.

We used a parse tree generated by a parser to detect the noun phrases in a sentence. Since the parse tree was not always correct, the number of definitions detected as noun phrases was reduced. We show two examples of mistakes in the parse tree in Figure 3 and 4.

Sentence (target mathematical expression is MATH_20):
From our first assumption , the contribution of the MATH_19th source to pixel MATH_20 of the MATH_21th channel image will be:…
Definition of MATH_20: pixel

Parse tree:

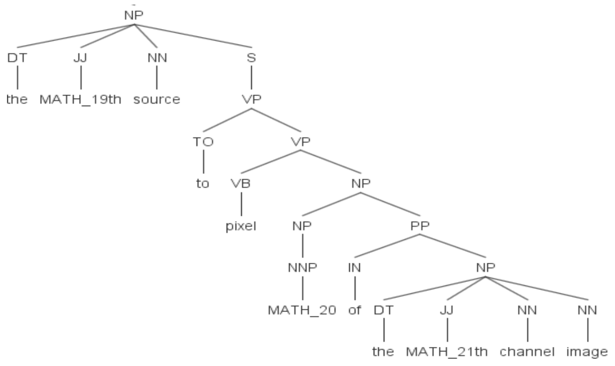


Figure 3. Undetected definition in incorrect parse tree

Sentence (target mathematical expression is MATH_95):
Let, MATH_95 be the set of pixel coordinates depicted in block MATH_96 and meanwhile they do not comprise pixel of the cavity.
Definition of Math_95: the set of pixel coordinates depicted in block MATH_96

Parse tree:

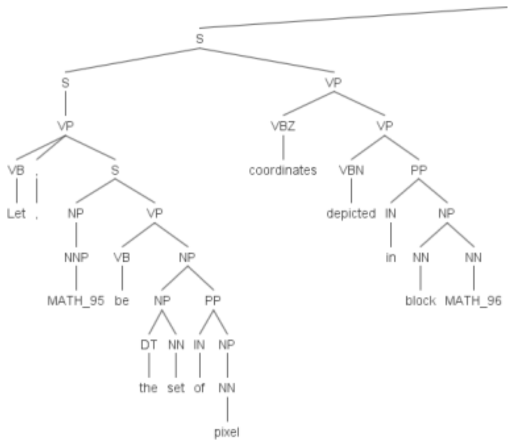


Figure 4. Detected short definition in incorrect parse tree

Figure 3 shows a mistake by the parser in parsing the "to pixel MATH_20 of the MATH_21th channel image" part. This part is should not be a verb phrase (VP), but should be a prepositional phrase (PP). Since it is parsed as VP, "pixel" is tagged as verb (VB). Therefore, it is not detected as a definition of MATH_20. In the correct case, the "pixel" must be a noun (NN) and part of the noun phrase (NP) "pixel MATH_20 of the MATH_21th channel image".

Figure 4 shows another mistake in the parse tree. The right subtree containing "coordinates depicted in block MATH_96" must be in the same noun phrase (NP) with a subtree containing "the set of pixel". Since the two parts of the definition are in different phrases, the complete definition of MATH_95 cannot be detected. In the strict matching measurement, MATH_95 is considered to be a mathematical expression without a definition. However, in the soft matching measurement, the short version of definition, which is "the set", can be detected as a noun phrase. Thus, MATH_95 is considered to have a definition.

7.2 Performance Measures

We considered two matching scenarios in this experiment: strict matching and soft matching. In strict matching, all of the extracted definitions must be exactly the same as the reference definitions from the annotation result. However, in soft matching, we also consider extracted definitions that are part of the reference definitions. Thus, we used three metrics to measure the performance of every method: precision, recall, and F1-score.

$$\boldsymbol{Precision} = \frac{\#math\ expressions\ with\ correctly\ extracted\ definitions}{\#math\ expressions\ with\ extracted\ definitions}$$

$$\boldsymbol{Recall} = \frac{\#math\ expressions\ with\ correctly\ extracted\ definitions}{\#math\ expressions\ with\ definitions}$$

$$\boldsymbol{F1} = 2 \times \frac{precision \times recall}{precision + recall}$$

7.3 Experiment Result

To get a more generalized evaluation, we combined ten training papers and two test papers and then performed 6-fold cross-validation on them. Table 14 shows the results for the three methods from the cross-validation.

Table 14. Performance Measures for 6-fold Cross-Validation

	Strict Matching			Soft Matching		
	P	R	F1	P	R	F1
Baseline Method						
Development	45.10	27.38	34.07	86.27	52.38	65.19
Test	33.04	15.12	20.68	81.94	37.42	51.22
Pattern Matching Method						
Development	26.92	33.33	29.79	48.08	59.52	53.19
Test	25.53	20.84	22.91	55.41	44.80	49.44
Machine Learning Based Method						
Development	92.22	33.13	48.61	96.90	45.27	61.70
Test	73.60	30.09	42.46	80.08	40.30	53.29

We can use Table 14 to compare the performances of the baseline and pattern matching. Certainly, the recall (R) of pattern matching is better than the baseline, since pattern 7 used in pattern matching is an extension of the baseline method. However, the improvement in recall had by pattern matching is not followed by the precision (P). Many sentence patterns led to there being many attempts in detecting the definitions. Since not every definition extracted from the sentence patterns is the actual one, many false attempts occurred in the pattern matching based method.

The cross-validation results show that the machine learning method with the strict matching measurement outperforms the baseline and pattern matching methods. On the test data set, it gives better precision, recall, and F1-score. For the soft matching measurement for the test data set, the machine learning method gives a better F1-score compared with the two previous methods. By looking at data statistics in Table 12, the upper limit for the machine learning method is lower than those of the baseline and pattern matching methods. Therefore, the machine learning's recall performance is not substantially better than these two methods, especially in the soft matching measurement.

8. Conclusion

We discussed ways of extracting definitions of mathematical expressions in scientific papers. The baseline method detects definitions by considering the surface texts preceding the target mathematical expression and their POS tag. The pattern matching based method detects definitions by considering only surface texts of sentences containing the target mathematical expression. In the machine learning based method, definitions are detected by firstly detecting noun phrases. Since we assume that the definition is always a noun phrase, these noun phrases are considered to be definition candidates.

Our approach using machine learning based showed potential in extracting such definitions in an experiment. Its performance in strict matching measurement was better than those of the naïve baseline or pattern matching based method. However, there is still room for improving this method. Several possible improvements might be: (1) using a parser that is more suitable for parsing sentences that contain mathematical expressions, (2) implementing a pattern generator automatically and using the extracted patterns as one of the features of machine learning, (3) refining the annotation concept to include more information about definitions, such as short versions of definitions, and (4) enabling extraction of fragmented definitions.

Acknowledgement

We would like to thank Springer for giving their permission for using the latex source of papers from the International Journal on Document Analysis and Recognition (IJDAR) and Graphs-and-Combinatorics in this research.

References

[Cramer 08] Irene Cramer, Finding General-Language Definitions in Corpora: Conceptual Design and Annotation, KONVENS 2008 - Ergänzungsband Textressourcen und lexikalisches Wissen, Zentrum Sprache BBAW Berlin, (2008).

[Jeschke 07] Sabina Jeschke, Marc Wilke, Marie Blanke, Nicole Natho, Olivier Pfeiffer, Information extraction from mathematical texts by means of natural language processing techniques, ACM Multimedia EMME Workshop 2007, pp. 109-114, (2007).

[Kohlhase 09] Michael Kohlhase, An Open Markup Format for Mathematical Documents OMDoc [Version 1.2], Springer Verlag. Lecture Notes in Artifical Intelligence, (2009).

[LateXMLPresentation] LateXML A LaTeX to XML Converter. http://dlmf.nist.gov/LaTeXML/.

[MathML] MathML Presentation Markup, http://www.w3.org/TR/MathML2/chapter3.html

[Minh-Quoc 10] Minh-Quoc Nghiem, Keisuke Yokoi, Yuichiroh Matsubayashi, Akiko Aizawa, Mining coreference relations between formulas and texts using Wikipedia, The Second International Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), (2010).

[Okazaki 07] Naoaki Okazaki. CRFsuite: a fast implementation of Conditional Random Fields (CRF). http://www.chokkan.org/software/crfsuite. (2007)

[Trzeciak 95] J. Trzeciak, Writing Mathematical Papers in English: A Practical Guide, Warsawa: European Mathematical Society, (1995).

[Wolska 04] Magdalena Wolska and Ivana Kruijff-Korbayova, Building a dependency-based grammar for parsing informal mathematical discourse, IJCNLP-04 Workshop, (2004).

[Yokoi 11] Keisuke Yokoi, Minh-Quoc Nghiem, Yuichiroh Matsubayashi, Akiko Aizawa, Contextual Analysis of Mathematical Expressions for Advanced Mathematical Search, 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing2011), (2011).

[Zinn 04] Claus Zinn, Understanding Informal Mathematical Discourse (Dissertation), (2004).